



Høgskolen i Telemark

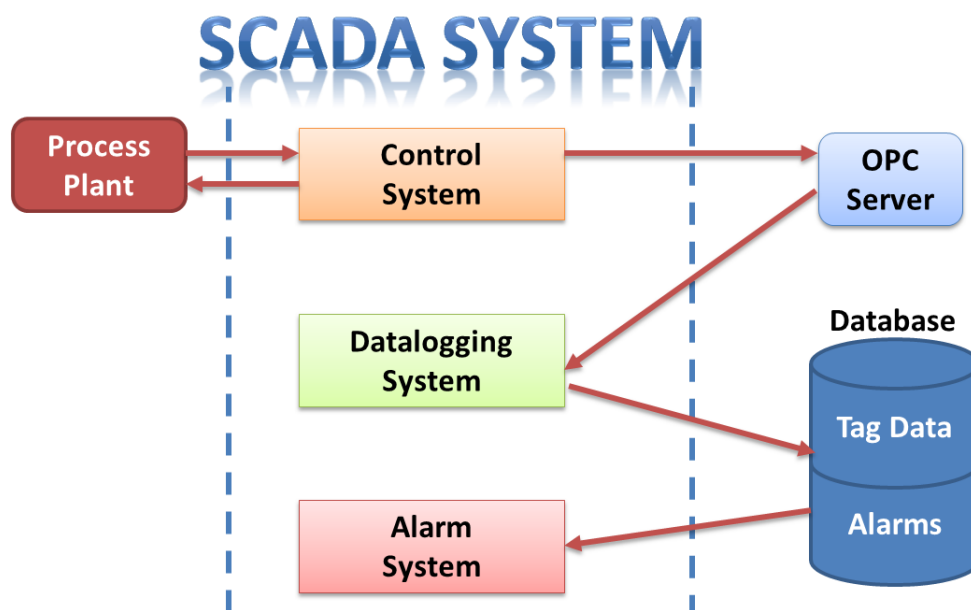
Telemark University College

Department of Electrical Engineering, Information Technology and Cybernetics



SCADA, OPC and Database Systems

HANS-PETTER HALVORSEN, 2012.08.20



Preface

In this Lab we will design our own **SCADA** system from scratch. We will use **SQL Server** to store our Measurement data we get from a **MatrikonOPC Server**. **LabVIEW** have built-in features for communication with **OPC**.

We will use LabVIEW to create our **GUI/HMI**. As part of the SCADA system we will create an **Alarm System** as well.

Go to the following website for more background information about the Lab Work:

→ <http://home.hit.no/~hansha/?lab=scada>

The purpose with this lab is to learn practical issues about the following topics in Industrial IT:

- SCADA systems
- Control Systems
- Human-machine Interface (HMI)
- Alarm Systems
- OPC (OPC Servers and OPC Clients)
- Database Systems, such as Microsoft SQL Server
- Structured Query Language (SQL)
- Open Database Connectivity (ODBC)
- Programming (LabVIEW, SQL, etc.)

Supervisory Control and Data Acquisition (SCADA) is an industrial control system monitoring and controlling a process.

A SCADA system consists of different modules/systems that together make the SCADA system, e.g.:

- Database System
- Management System
- Control System
- Datalogging System
- Monitoring System
- Alarm System
- Report System

In this assignment we will implement our own “SCADA-system” from scratch in LabVIEW. The Database will be created in SQL Server.

A SCADA system consists of many modules, but our main focus will be to create the Database system and to create a simple Control System and Alarm System.

Table of Contents

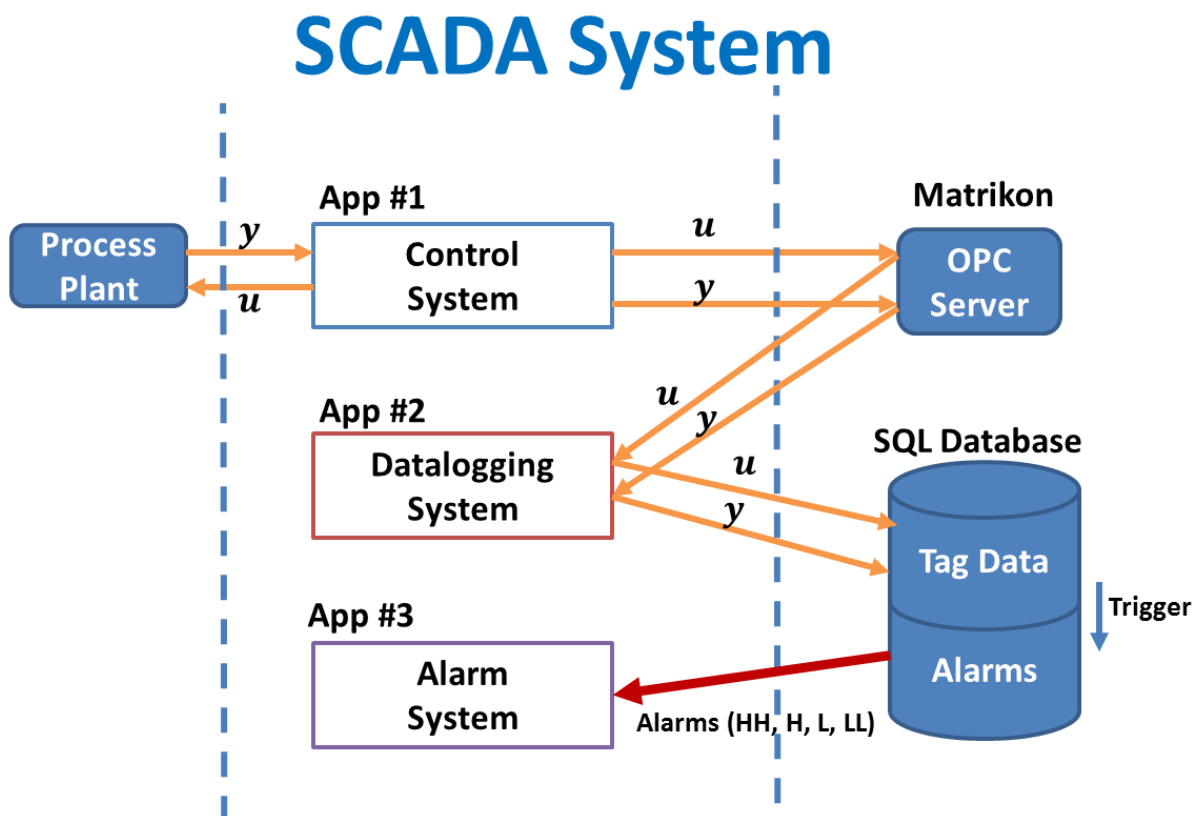
Preface.....	ii
Table of Contents	iv
1 Introduction.....	1
1.1 SCADA.....	1
1.2 Process Plant	2
2 Introduction to OPC.....	3
2.1 MatrikonOPC Simulation Server.....	3
2.2 MatrikonOPC Explorer.....	3
2.3 Write Data	4
2.4 OPC Client in LabVIEW.....	4
2.5 Read OPC Data.....	5
2.6 Write OPC Data.....	6
3 Database Communication in LabVIEW	7
3.1 ODBC.....	7
3.2 SQL Toolkit.....	7
4 Database System	10
4.1 Introduction.....	10
4.2 Tasks	12
Task 1: Database System	12
5 Control & Datalogging System.....	13
5.1 Introduction.....	13
5.2 Tasks	13
Task 2: Control System	13

Task 3: Datalogging.....	15
6 Alarm System.....	16
6.1 Introduction.....	16
6.2 Tasks.....	17
Task 4: Alarm Logging.....	17
Task 5: Alarm List.....	18

1 Introduction

1.1 SCADA

In this assignment we will create a small-scale SCADA application.



We will focus on creating 3 different systems as shown in the figure. The 3 different systems will be:

1. Control System
2. Datalogging System
3. Alarm System

Note! These 3 systems shall be implemented as separate applications that in theory can be running on different computers.

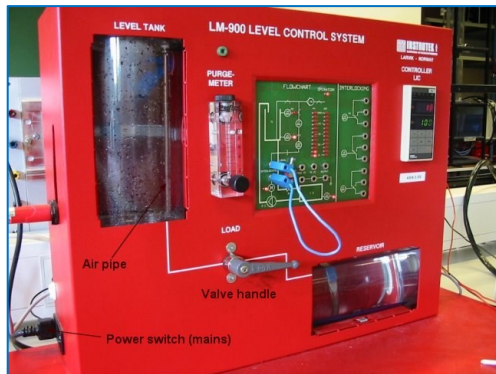
We will use LabVIEW to create these systems, but if you want an extra challenge you can use, e.g., C# to implement one or more of these systems.

1.2 Process Plant

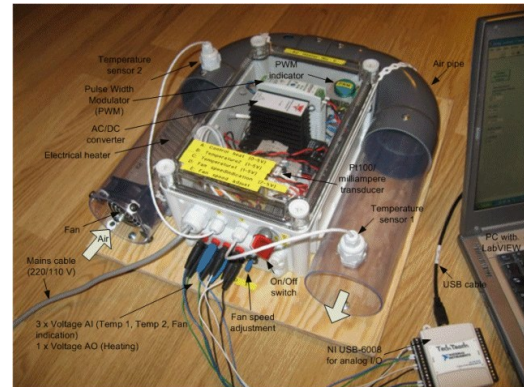
In this Assignment we will use a mathematical model of either the **Level Tank** or the **Air Heater** system. Because focus in this assignment is not Data Acquisition we will only use a mathematical model in this assignment.

Below we see the Lab Equipment available for this assignment:

Level Tank



Air Heater



Documents of how to use the Level Tank/Air Heater and the USB-6008 DAQ device is available from <http://home.hit.no/~hansha>.

- **Level Tank:** <http://home.hit.no/~hansha/?equipment=leveltank>
- **Air Heater:** <http://home.hit.no/~hansha/?equipment=airheater>
- **USB-6008:** <http://home.hit.no/~hansha/?equipment=usb6008>

2 Introduction to OPC

Here is a short introduction of how to communicate with OPC in LabVIEW. For more information you should read the Tutorial [“OPC and Real-Time Systems in LabVIEW”](#).

2.1 MatrikonOPC Simulation Server

In this Lab Work you will use the “**MatrikonOPC Server for Simulation**” and “**MatrikonOPC Explorer**”.

For more information about this software, see the [“OPC and Real-Time Systems in LabVIEW”](#) Tutorial.

Install the software MatrikonOPC Simulation Server (if you haven't already done it). Link to download page is available from the homepage of this Lab Work.

Please read the documents available for the MatrikonOPC Server.



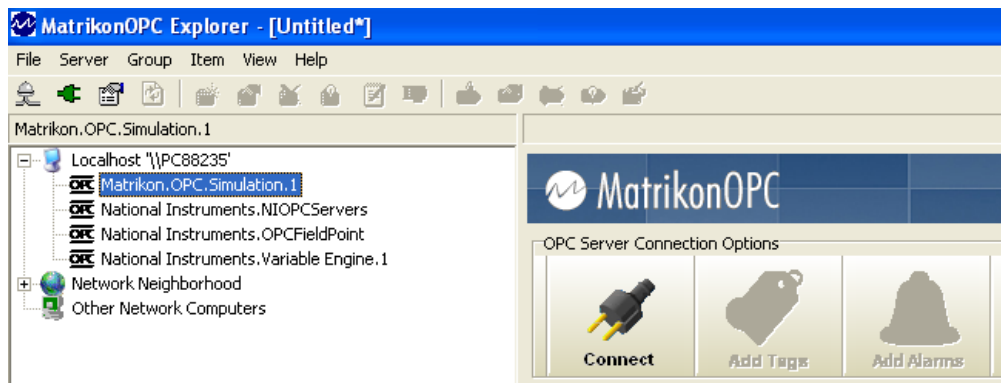
Go through the Quick Start Guides and **User Manuals**:

- MatrikonOPC Server for Simulation Quick Start Guide
- MatrikonOPC Server for Simulation User Manual
- MatrikonOPC Explorer Quick Start Guide
- MatrikonOPC Explorer User Manual

They are available from the Start menu after the installation of the software.

2.2 MatrikonOPC Explorer

The MatrikonOPC Explorer is a test tool you can use to get data, etc. from different OPC Servers.

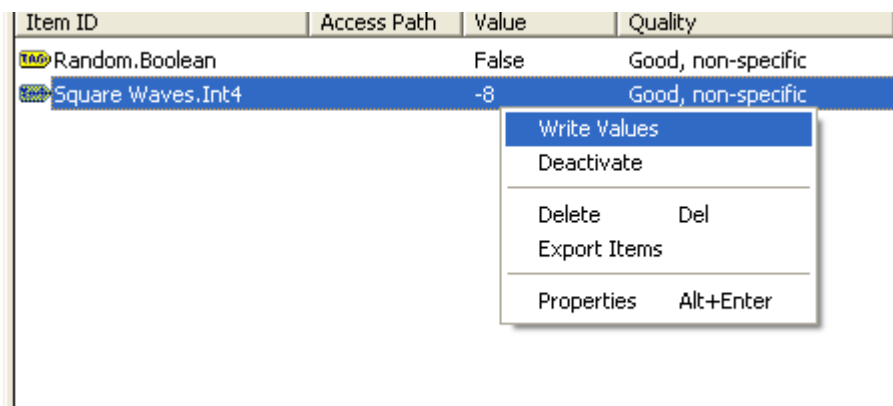


Open the **MatrikonOPC Explorer** and connect to the MarikonOPC Server for Simulation.

MatrikonOPC Explorer continually updates the display for that group with real-time data. The data values appear in the list view on the right-hand side of the OPC Explorer window. Each item is listed along with its item ID, access path, active state, value, quality and timestamp.

2.3 Write Data

You may also write data to the OPC server.

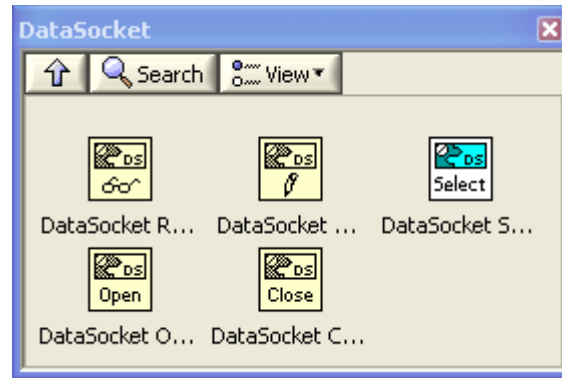


Note! You cannot write to items that are set to be “read-only”!

2.4 OPC Client in LabVIEW



In order to communicate with an OPC server in LabVIEW you use the DataSocket VIs.



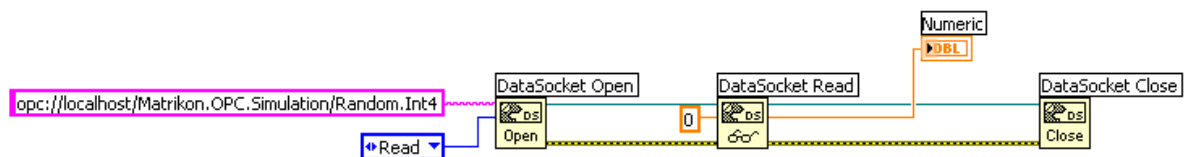
2.5 Read OPC Data



DataSocket Read

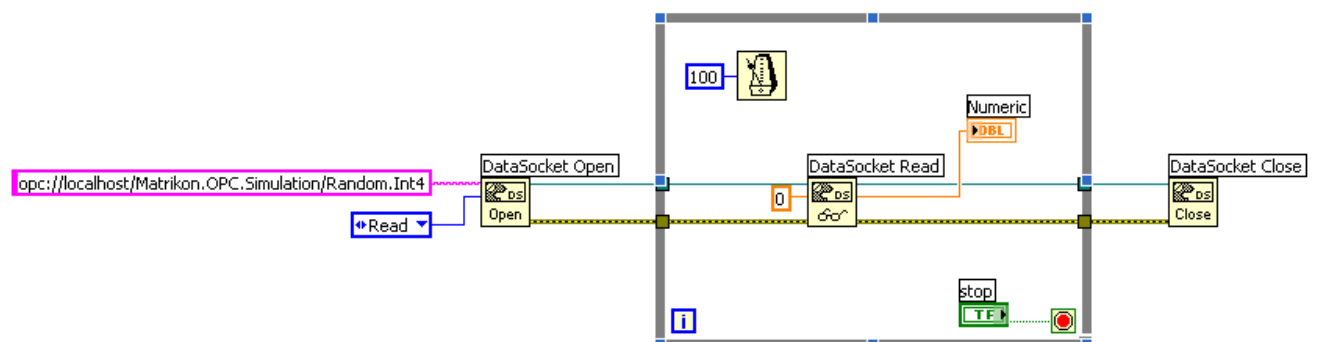
You use the DataSocket Read VI for reading OPC data.

Example:



Use a while loop in order to read the OPC Data continuously.

Example:



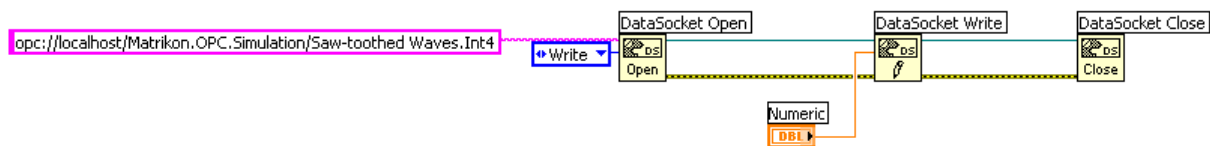
2.6 Write OPC Data



DataSocket Write

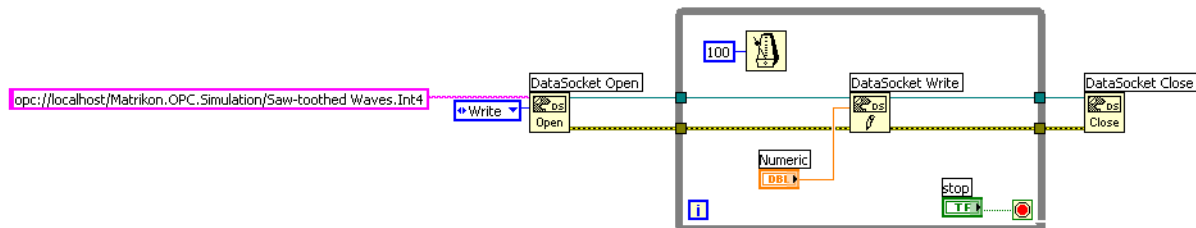
You use the DataSocket Write VI for reading OPC data.

Example:



Use a while loop in order to write to the OPC server continuously.

Example:

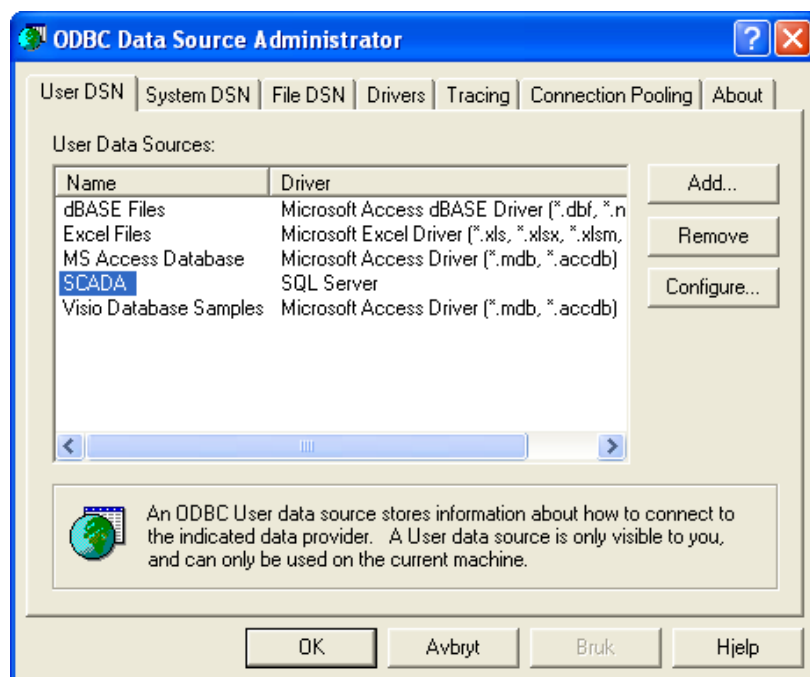


3 Database Communication in LabVIEW

Here is a short introduction of how to communicate with a Database in LabVIEW. For more information you should read the Tutorial "[Database Communication in LabVIEW](#)".

3.1 ODBC

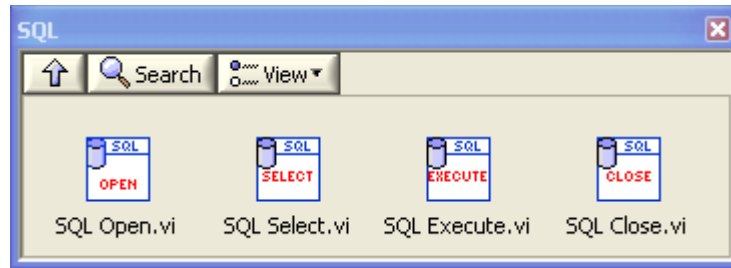
You may want to set up an ODBC Data Source, which is a link to your Database system.



3.2 SQL Toolkit

I have made a simple and easy to-use SQL Toolkit. The SQL Toolkit is available for download from the homepage of this lab.

When you install this toolkit, the following palette is available in LabVIEW:



The SQL Toolkit contains the following VIs:



"SQL Open.vi" - This VI open a connection to the database specified in the Connection string. The connection string may be as follows:



"SQL Select.vi" -This VI get data from the database specified in the SQL Query. The output is a 2D string array with data.



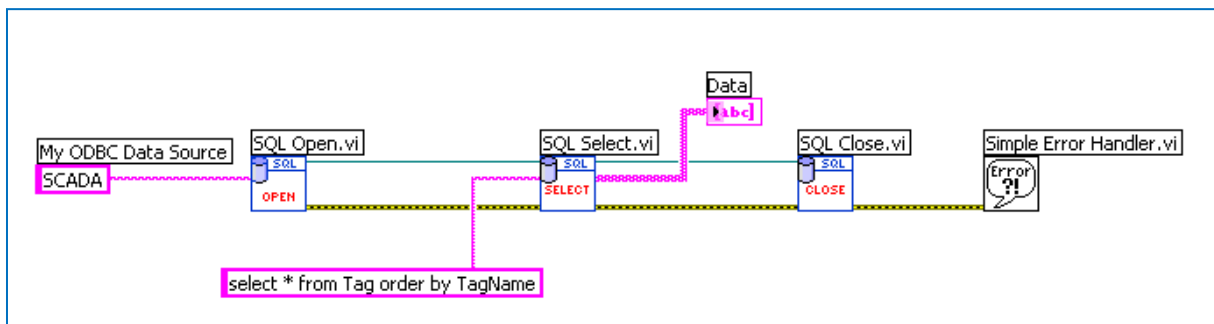
"SQL Execute.vi" - This VI executes a Query with no return Data, e.g., an INSERT statement



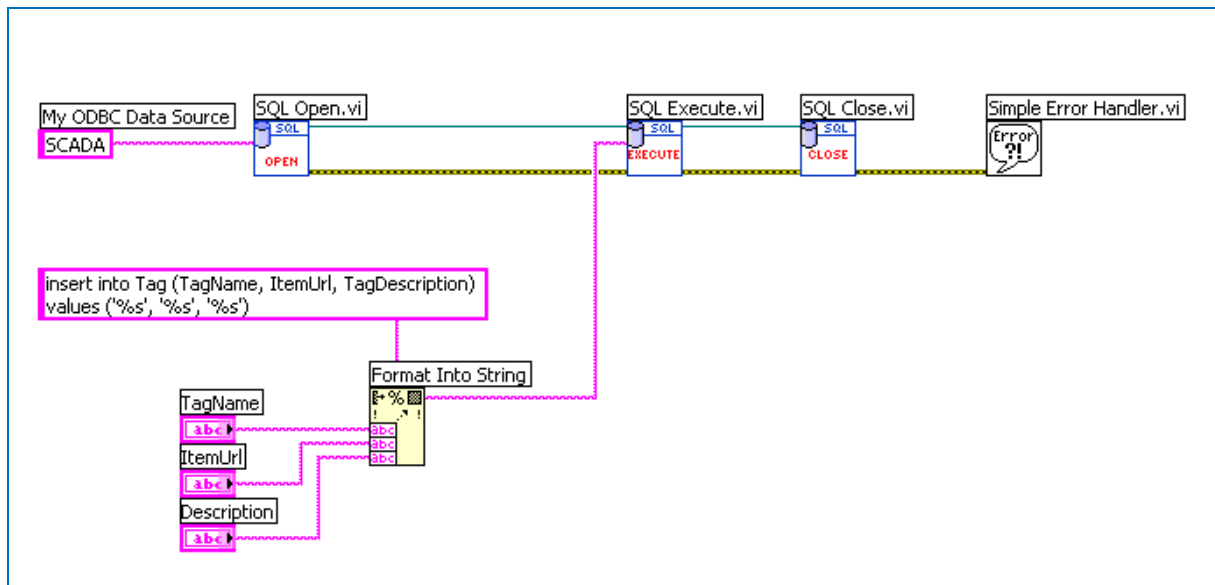
"SQL Close.vi" - This VI Close the connection to the database opened by "SQL Open.vi"

The VIS can be used for both writing data to the database and reading data from the database. Below we see some simple examples:

Reading Data from the Database:



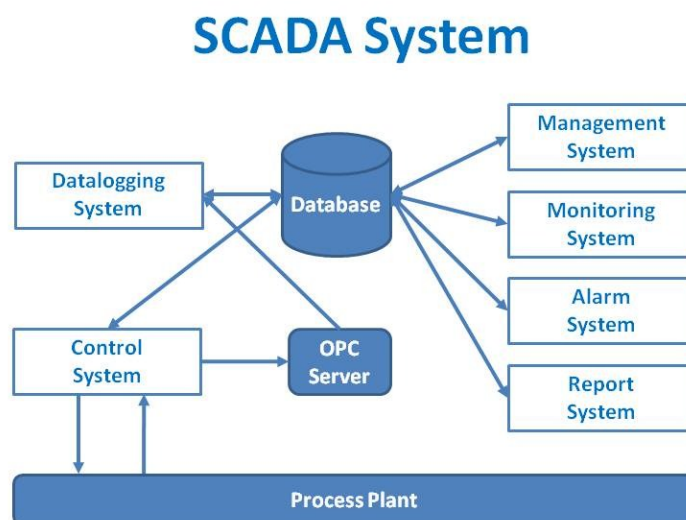
Writing Data to the Database:



4 Database System

4.1 Introduction

In this Task we will design and create the Database System.



In this Assignment we will use Microsoft SQL Server as a repository for our data. We will use Microsoft Visio to design our database system.

Note! When defining your tables and columns, so-called “Pascal notation” is recommended, i.e., each word should start with an uppercase, e.g., “TagId”, “ItemUrl”, etc. You should also avoid using plural and only stick to singular, i.e, use “Tag” and not use “Tags”.

We can divide the tables needed to be created in different parts, such as:

- Tag Configuration
- Tag Data
- Alarm Configuration
- Alarm Data

- etc

Example:**Tag Configuration:**

The TAG table could e.g. have the following columns:

- TagId (int, Primary Key, Identity)
- TagName (varchar)
- ItemId (varchar)
- ItemUrl (varchar)
- Description (varchar)
- etc.

Create more tables and columns if necessary.

Tag Data:

Create one or more tables used for logging the Tag Values into the Database.

The following fields are important for an OPC Item:

- ItemId
- Value
- Timestamp
- Status (e.g., "Active", "Not Active")
- Quality (e.g., "Good", "Bad")
- etc.

Alarm Configuration & Alarm Data:

Important fields in an alarm handling system could be:

- Alarm id
- Activation time
- Acknowledge time
- Type of alarm
- Tag
- Alarm limit
- Text description
- etc.

The alarm system will be responsible for the warnings and the alarms in a monitoring and control system. An alarm system contains different **Alarms** and **Warnings** like:

- Timeout; no input from a sensor or another computer system within a specific amount of time,
- High High (HH) or Low Low (LL) alarm; a critical alarm condition,
- High (H) or Low alarm (L)
- I/O device errors
- System device errors
- etc.

Make sure your alarm tables and system can handle these kinds of alarms and warnings.

An Alarm System use different Alarm devices, such as

- Screen; display the alarms
- Keyboard; alarm operations
- Horn; indicate an active alarm, or security alarm
- Lamp; indicate an active alarm, or an active alarm by blinking and an acknowledge alarm by a steady light
- Printer; logging of the alarm states
- SMS
- E-mail
- Etc.

Make use of one or more of these alarm devices in your Alarm Handling and Management System.

You should create one or more tables that set up the alarm configuration for a specific tag and one or more tables that store the alarms being generated (even if you are not selecting the optional Alarm System task, you should design the tables for the Alarm Handling).

4.2 Tasks

Task 1: Database System



Design and create your tables for the SCADA system (as illustrated above).

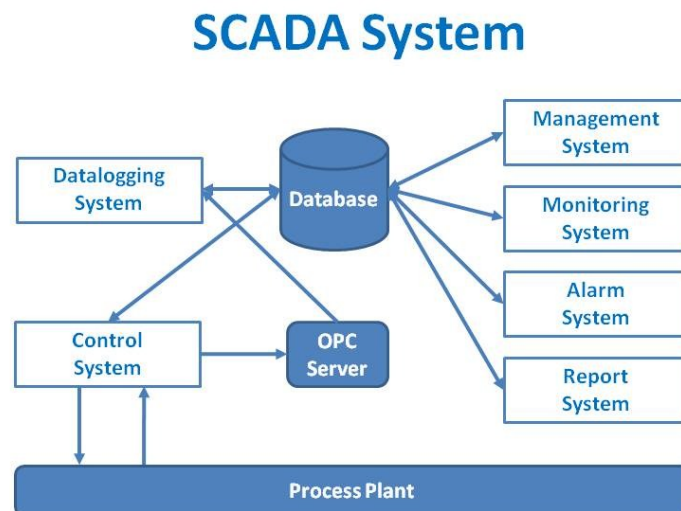
Tip! Create a SQL Script that inserts these tables into your database.

[End of Task]

5 Control & Datalogging System

5.1 Introduction

In this Task we will create a Control System using, e.g., the built-in PID function in LabVIEW. The process data (Process Value, PV and Manipulated Value, MV) should be stored in the OPC Server.



5.2 Tasks

Task 2: Control System



Create an Application in LabVIEW that controls the level in the Level Tank or the Temperature in an Air Heater. Use standard PI(D) control.

→ You need to create a HMI (Human Machine Interface) for your application.

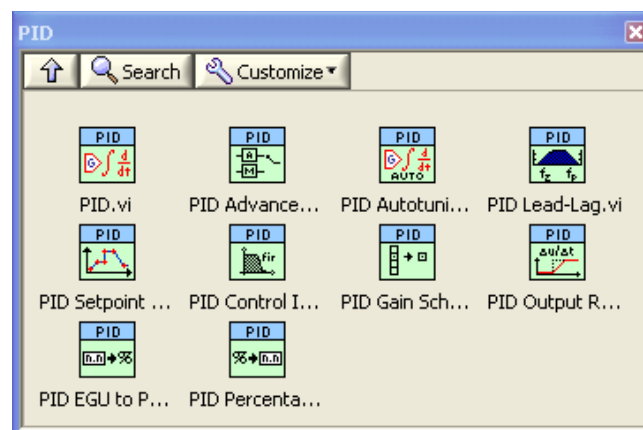
→ The process data should be sent to the OPC Server. You should write the Control Signal (u) and the Process Value (y) to the OPC Server, so you need to establish these items in the OPC Server in advance.

Note! Since “MatrikonOPC Server for Simulation” is just a “demo” OPC Server, you cannot create your own OPC items, you have to use on or more of the existing items that comes with the product.

Tip! Use e.g., the OPC items “Bucket Brigade.Real4” and Bucket Brigade.Real8 for this.

PID Control:

You may want to use, e.g., the built-in PID functionality in LabVIEW (PID Advanced.vi is recommended):

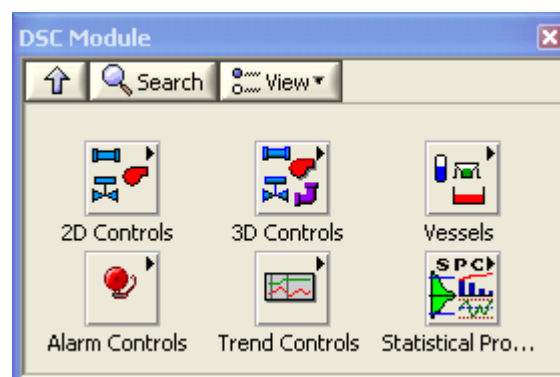


Note! You don't need to use the real process, it is enough using a simulated model in this Task.

HMI:

Your HMI should consist of water tanks, pumps, pipes, etc.

You may want to use the LabVIEW DSC (Datalogging and Supervisory Control) palette in LabVIEW in order to create your own HMI:

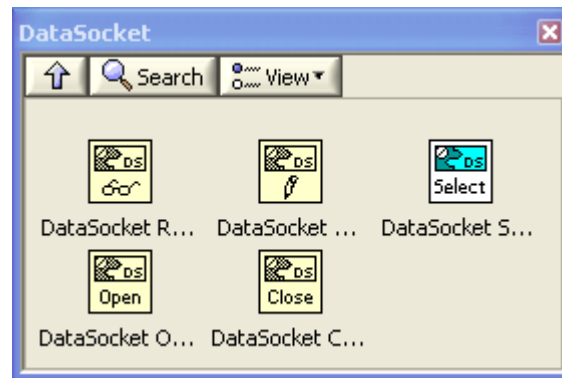


Alarms:

As part of the HMI you should illustrate in the User Interface when the process value is below the alarm limit (Low Alarm) or the process value is above the alarm limit (High Alarm).

OPC:

Use the **DataSocket** VIs in LabVIEW for communication with the OPC Server.



Use the **MatrikonOPC Server for Simulation**.

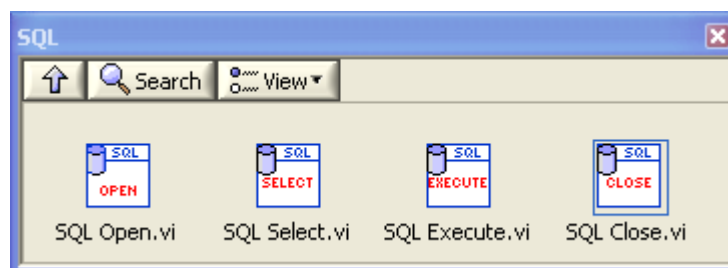
Note! We will use LabVIEW to create this application, but if you want an extra challenge you can use, e.g., C# to implement the application.

[End of Task]

Task 3: Datalogging

Create a simple application in LabVIEW that reads the values from the OPC Server and store them in the SQL Server.

Download and use the “**SQL Toolkit**” available from the Homepage of this Lab Work.



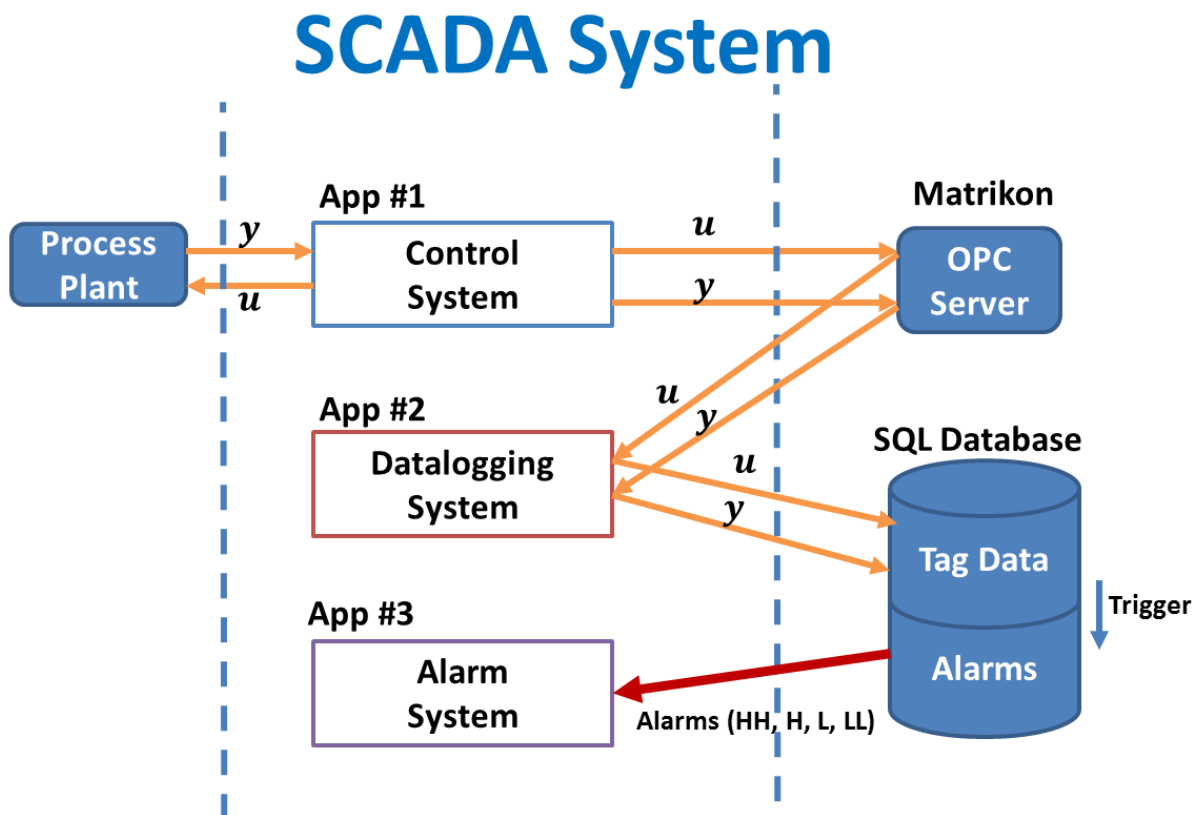
Note! We will use LabVIEW to create this application, but if you want an extra challenge you can use, e.g., C# to implement the application.

[End of Task]

6 Alarm System

6.1 Introduction

In this Task we will create an Alarm System. The Alarm System checks for Alarms and saves the Alarm information in your Database. In addition you should have a User Interface that shows the Alarms.

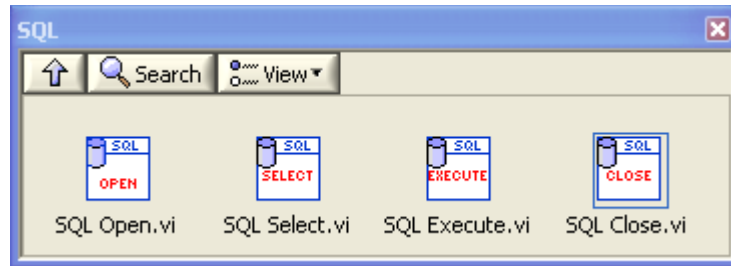


In this task you will create an Alarm Handling System in LabVIEW and use **Microsoft SQL Server** as the Alarm Repository.

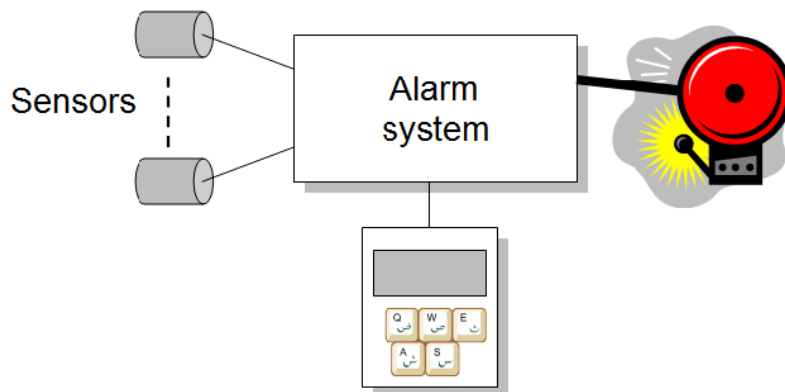


Please read the Tutorial "[Database Communication in LabVIEW](#)" before you start this task."

Download and use the "**SQL Toolkit**" available from the Homepage of this Lab Work.



In this part you will extend your system with **Alarm Handling**. You will need to create a system for logging alarms to a database and systems for viewing and operate on the alarms.



In an alarm system the use of colors are important, so make sure your system use colors according to standards for HMI and SCADA systems.

6.2 Tasks

Task 4: Alarm Logging



Create a system for Alarm Logging and to save these alarms in the database system.

Tip! You could implement Alarm Logging in your SCADA system by using a Database **Trigger** on the table that stores the Tag Data.

Start with checking, e.g., **Low Alarm** (process value is below a specified limit) and **High Alarm** (process value is above a specified limit). Extend the system with more alarm types if you want to.

[End of Task]

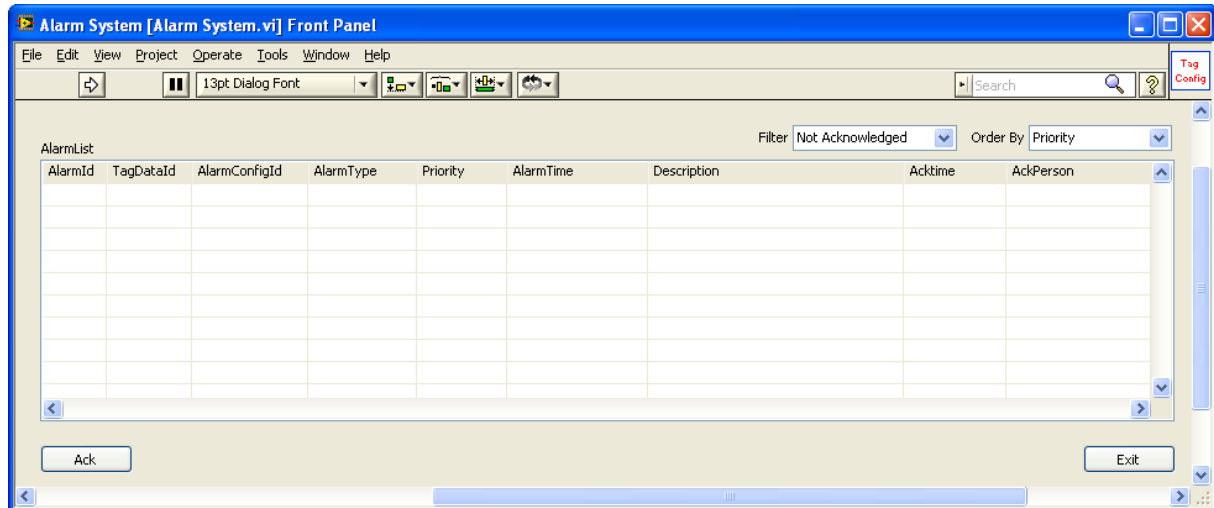
Task 5: Alarm List



Create an Alarm List application that shows all the alarms in the system.

The operator should be able to see the alarms and make some actions to these alarms, i.e. the operator should have the possibility to acknowledge alarms.

The program could look something like this (feel free to create your own User Interface):



Note! We will use LabVIEW to create this application, but if you want an extra challenge you can use, e.g., C# to implement the application. Creating the Alarm System as a web application is also a good idea, using e.g., ASP.NET and C#.

[End of Task]



Høgskolen i Telemark

Telemark University College

Faculty of Technology

Kjølnes Ring 56

N-3914 Porsgrunn, Norway

www.hit.no

Hans-Petter Halvorsen, M.Sc.

Telemark University College

Department of Electrical Engineering, Information Technology and Cybernetics

E-mail: hans.p.halvorsen@hit.no

Blog: <http://home.hit.no/~hansha/>

